

### REMARKS

Claims 1-9, 11-17, 19-38, 40-44, 46-50, 52-56, and 58-60 remain pending in the application. Favorable reconsideration is respectfully requested in view of the following remarks.

Claims 1-9, 11, 17, 19, 20-22, 32, 34, 38, 40, 44, 46, 50, 52, 56, 59, and 60 stand rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Childers et al. (US Patent 5,986,913) in view of Campbell et al. (US Patent 5,021,947). This rejection is respectfully traversed.

As explained in an earlier-filed Amendment, a processor in accordance with the invention is formed of a plurality of processing elements (PEs) adapted to receive an incoming stream of data packets, which may be of unpredictable length. There is no requirement for an incoming packet stream in which every packet is of the same fixed size. In the case of a small packet, it can be read into a single processing element along with its header. Where an incoming packet is large, it is "chopped up" into smaller fragments or "chunks" of substantially equal size and spread over as many PEs as are necessary to store it and process it. This is spelled out in the description and is specified in original claim 4. The product of packet size and packet rate is invariably constant (see the specification at page 7, lines 19-21), so there is no possibility of the incoming stream having large packets and a high packet rate at the same time.

Thus, each chunk is sent to a respective PE. If an incoming large data packet is not an integral multiple of the size of a chunk, there will be a residual chunk that is not the same size as the rest. This residual chunk will nevertheless be sent to a PE along with the others. It could be padded out with zeroes to make it the same size as all the other chunks but its length could simply be indicated in some way, such as in a header for that chunk. As indicated at page 59, lines 15-20 of the specification, the processor is matched to a constant line bandwidth rather than to any particular number of packets.

Various passages throughout the specification support these various characteristics, such as page 12, line 16; page 13, lines 6-9; page 22, lines 19-21; page 23, lines 26-32; page 24; page 51; and page 59. Claim 1 of course relies on these characteristics, and defines a data processing architecture "wherein the input device is operable to distribute whole data packets of potentially varying size across one or more processing elements such that the number of processing elements across which each whole data packet is distributed is dynamically

determined based at least in part on the size of the whole data packet.” As a result of this arrangement, small packets are distributed “whole” (i.e., an entire (small) packet is sent to a single PE), whereas a large packet is distributed “in parts” over a number of PEs so that each of those PEs stores only a part (a chunk) of the packet.

Turning now to specific aspects of the Office Action, the Office relies, in part, on Childers et al. as showing some of the features defined by independent claim 1. The Office acknowledges that the Childers et al. patent does not show the number of PEs being determined based on the size of the data packet as claimed, but relies on Campbell et al. as making up for these deficiencies. This reliance is unfounded at least because one of ordinary skill in the art at the time of Applicants' invention would not have found any motivation to combine the teachings of Childers et al. with those of Campbell et al. Moreover, even if one were to make such a combination, as now suggested by the Office, that combination would still fail to include at least a data processing architecture “wherein the input device is operable to distribute whole data packets of potentially varying size across one or more processing elements such that the number of processing elements across which each whole data packet is distributed is dynamically determined based at least in part on the size of the whole data packet,” as defined in claim 1.

To see why this is so, consider first the Childers et al. patent, which discloses a high-speed sense amplifier/ memory configuration resistant to voltage spikes. The practical embodiment is a serial video processor in which lines of video are encoded to create packets. Each line appears to constitute a packet. Each line is sampled to produce N samples, each of one word. The processor is a SIMD/RISC device consisting of a one-dimensional array of 1-bit PEs shown in Figure 1 as a parallel processor. Figure 2 represents a block diagram of a single PE of the processor. There are N number of PEs, which is the same number as there are pixels in the video line. Data for each pixel is handled by a respective PE. *Of particular relevance is that the packets are all of the same size.* (See, e.g., Childers at column 3, lines 24-28: “The ‘serial video’ aspects of SVP 10 derive from the fact that it is particularly suited for video processing, where discrete packets of incoming data, which have a uniform size, are input and output ....” Emphasis added.) This is in direct contradistinction to claim 1's recitation that the packets are “of potentially varying size.”

Moreover, there is considerable discussion in Childers et al. as to the number of bits for the PEs and the fact that there are up to N=1024 words x 40-bits of input data but it is

inevitable that the allocation of data to the PEs is based purely on *numbers* of pixels and *numbers* of PEs. The actual bit-values discussed in Childers et al. are conventional for video processing, so there is no particular significance in the choice of bit numbers.

It is clear from Childers et al. (e.g., see columns 5 to 6) that an object is to build a parallel processor to do a specific job. As a starting point at achieving this object, it is known in advance that (video) data packets of a certain predetermined and *unvarying* size will be handled. The processor is therefore designed to have as many PEs (N) as there are pixels/words in a video line, and each video line will be sampled N times. Further, the input registers are designed to have the same bandwidth (40 bits) as each word in the data input stream. It is therefore known that each and every PE will, time after time, process the same, predetermined amount of data. Nothing changes from line to line and from time to time. *The Childers et al. processor is decidedly static and cannot possibly be dynamic.* If a larger (or smaller) processor is needed, the whole architecture has to be redesigned for the new job. There is no prospect of living with the same processor but letting it dynamically allocate data to PEs on the basis of varying packet size.

In Childers et al., the processor has clearly been engineered to match the number of pixels with the number of PEs on a one-to-one basis. In contrast, a processor architecture in accordance with claim 1 distributes whole data packets "such that the number of processing elements across which each whole data packet is distributed is dynamically determined based at least in part on the size of the whole data packet." That is, Applicants' processor does not require that the size of packets in the incoming data stream be static and uniform, but instead automatically subdivides a larger packet into fragments, which are then allocated to as many PEs as are necessary to store it for a given bandwidth. It is not coincidence that there are as many PEs in Childers et al. as there are samples in a line or packet. In contrast, Applicants' allocation is based on data bandwidth/packet size as opposed to pixel number.

Moreover, it is respectfully contended that Childers et al. is not an appropriate document on which to base the rejection. One of ordinary skill in the art would surely be looking for prior art specifically directed at parallel processors for handling packets of varying size. Childers et al. is not such a reference, first because Childers et al. does not handle packets of varying size. Further, Childers et al. is concerned with a memory structure. The description of the processor in Childers et al. is merely the background scenery against which the featured memory structure is painted. It is most unlikely that the person of

ordinary skill in the art would have started with Childers et al. as a basis for creating a parallel processor architecture that would solve the problem of packet allocation based on packet size.

Nonetheless, even with Childers et al. as a starting point, it is hard to understand the relevance of Campbell et al. to the issue of obviousness. The Office refers in paragraph 4 only to Figure 15 of Campbell as motivation for one of ordinary skill in the art to consider making changes to Childers et al. in order to achieve Applicants' invention. Figure 15 merely illustrates a correlation between packet length and PE number. However, it is no surprise that the more PEs there are, the larger the number of locations for processing data from a given packet, provided of course the packet is distributed over all of the PEs. A very simple analogy would be a visit to a post office or bank. The more tellers there are behind the counter, the shorter the queue at each. Looking at it another way, a customer with a number of transactions to conduct could shorten his visit if he spread the transactions out over a number of different tellers.

However, what Campbell et al. singularly fails to teach is a dynamic approach, in which a processor allocates packets of unpredictable size to a number of PEs. In Applicants' invention, if a packet is too large to be handled whole by a single PE, it is chopped up and chunks of it are distributed to individual PEs. If a packet is small enough, it can be allocated to a single PE. Applicants' processor does not need to know in advance what size packets are arriving in an input data stream because it is intelligent enough to decide for itself how large the packets are and the capabilities of each PE, so it can distribute the whole packet or chunks of it to respective PEs in the processor. To quote from page 59, lines 15-18 of Applicants' specification, "Each processor handles a batch of packets sufficient to fill the local memories of its PEs. In effect it consumes a near constant amount of line bandwidth per processing phase, rather than a constant number of packets." Campbell et al. provides absolutely no incentive to do other than load one packet across all the PEs in the processor, whereas in Applicants' invention, some PEs may remain empty.

Campbell et al. states in column 1 at lines 62-68 that arrays are inflexible, confirming Applicants' argument. Campbell et al. is a wholly specific design to handle data flow in graphical form and utilizing a very specific high level programming language. Column 11 of Campbell et al., from line 12 onwards, discusses the distribution of parallel data flow graph information to PEs. This is acknowledged in column 13, lines 2-5 as involving no innovative

techniques. Significantly, Campbell et al. states categorically in column 14 at lines 23-25 that static allocation is preferred. If ever there was one, this is a clear teaching that would *deter* one of ordinary skill in the art from considering Campbell et al. as providing a motivation for exploring a dynamic approach in the Childers et al. architecture. In the allocation of graph data to PEs in Campbell et al., there is no mention of packet size. On the contrary, allocation is based on a combination of three heuristic cost evaluations, as set out in columns 15 and 16, namely communication cost, array-access cost and parallel processing cost. In this regard, the actual physical location of PEs relative to one another contributes to this evaluation. This heuristic approach applies also to the global allocation discussed in Campbell et al.

In view of the foregoing, it is apparent that Childers et al. is not an appropriate starting point for the evaluation of obviousness of Applicants' invention, and that Campbell et al. fails to provide any teaching that would have motivated one of ordinary skill in the art, at the time of Applicants' invention, to consider Campbell et al. as relevant to Childers et al. in the context of Applicants' architecture as defined by independent claim 1.

Claims 2-9, 11, 17, 19, 20-22, 32, 34, 38, 44, 46, 50, 52, 56, 59, and 60 variously depend from claim 1, and therefore define at least the same patentable subject matter discussed above.

For at least the foregoing reasons, it is respectfully contended that claims 1-9, 11, 17, 19, 20-22, 32, 34, 38, 40, 44, 46, 50, 52, 56, 59, and 60 are patentable over the prior art of record. Accordingly, it is respectfully requested that the rejection of these claims under 35 U.S.C. §103(a) be withdrawn.

Claims 12, 13, 35, 41, 47, and 53 again stand rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Childers et al. in view of Gove et al. (US Patent 5,371,896). This rejection is respectfully traversed.

Claim 12 depends from claim 1, and thereby incorporates all of the features of that base claim. Claim 12 is therefore patentably distinguishable over Childers et al. for at least the reasons set forth above. It is noted that the Office acknowledges, in numbered paragraph 4, that "Childers did not specifically show the number of PEs was determined based on the size of the data packet as claimed."

Gove et al. also fails to disclose an input device "operable to distribute whole data packets of potentially varying size across one or more processing elements such that the

number of processing elements across which each whole data packet is distributed is dynamically determined based at least in part on the size of the whole data packet,” and therefore fails to make up for the deficiencies of Childers et al. Because these features would be lacking in any combination of Childers et al. with Gove et al., such a combination would fail to support a *prima facie* case of obviousness.

Moreover, as explained in Applicants' previously filed response, the Office's reliance on Gove et al. as disclosing claim 12's recitation of a data processing architecture “wherein at least one processing element is operable to enter a standby mode of operation in dependence upon data received by that processing element” is unfounded. Applicants' claim 12 defines the processing element as being operable to enter a standby mode of operation in dependence upon data received by that processing element.” (Emphasis added.) (For example, whether a given one of the processing elements is activated or idle may depend on whether that processing element contains the packet header. See, e.g., specification at page 12, lines 15-22.) As explained in Applicants' previously-filed remarks, Gove et al. disclose a processor switchable between SIMD and MIMD using a cross-bar switch interconnecting various PEs and memory to change the combination of distributed/shared memory. There are several parallel processors interconnected with the memory banks. When there is contention (i.e., simultaneous accesses to RAM by any two system devices) a “SIMD pause” signal is routed to pause *all* PEs (see Gove et al. Fig. 30). *The pause is therefore not dependent on data, as required by the Applicants' claim 12*, but is instead dependent on the occurrence of contention in a separate unit. The standby mode required of Applicants' claim is therefore not triggered in the way disclosed by Gove et al.

Claims 13, 35, 41, 47, and 53 depend from claim 12, and are therefore patentably distinguishable over the Gove et al. patent for at least the same reasons as those set forth above.

In view of the foregoing, it is respectfully asserted that claims 12, 13, 25, 41, 47, and 53 are patentable over the prior art of record. Accordingly, it is respectfully requested that the rejection of these claims under 35 U.S.C. §§ 103(a) be withdrawn.

Claims 23-31 and 58 again stand rejected under 35 U.S.C. §§ 102(a) and (b) as allegedly being anticipated by Horst (US Patent 5,404,550). This rejection is respectfully traversed.

This rejection was previously made in the Office Action dated November 17, 2004 and Applicants responded on April 18, 2005 with full argument and reasoning why Horst was not considered to anticipate claim 23. The Office appears to have overlooked the arguments raised in relation to claim 23 in that response, or at least has not given any indication why those arguments are not considered to be persuasive. Since Applicants are still strongly of the opinion that the rejection is not sustainable, the salient points raised in the previous response are repeated here. If the Office still considers the claim rejection to be valid, further explanation regarding why the rejection is considered to be well-founded would be greatly appreciated.

Claim 23 defines an input/output system for transferring data to and from a plurality of processing elements arranged in a single instruction multiple data (SIMD) array, the system being operable to transfer data packets of different sizes to respective ones of the processing elements in the array.

To a large extent, this claim is tied in with claim 1, in that it comprehends the same overall concept of a processor comprising an array of PEs, wherein the PEs are operable with data packets of variable size. Applicants are here claiming the aspect that the input/output system transfers data packets of different sizes to respective PEs; that is, whole packets are delivered to respective PEs, one packet per one PE. It should be appreciated that claim 23 provides a mechanism for distributing packets to PEs in a SIMD array and is not concerned with transfer of data between PEs in the array. What happens to data packets once they have been distributed to PEs across the array is immaterial to the I/O aspect specifically claimed in claim 23.

This aspect has a different slant from that in claims 1 and 10, in which a single packet is subdivided into a plurality of fragments and the fragments sent to respective PEs of a processor array. Instead, claim 23 involves an I/O system for transferring data to and from PEs in a SIMD array such that data packets of different sizes are sent to respective ones of the PEs in the array. In other words, the PEs are large enough to accept whole packets.

The Office relies on Horst as being relevant to the novelty of claim 23. This reliance is unfounded for a number of reasons. For one thing, Horst discloses inter-processor networks as opposed to I/O systems. The Office alleges that Horst discloses an input and output system for transferring requests from the processing elements to the hardware

accelerator, but Applicants respectfully disagree that this makes Horst's arrangement "an I/O system."

Horst discloses a mechanism for routing message blocks *between* PEs. If there is no direct path, then the messages are routed through intervening PEs to their destination. Neither of these is explicitly for I/O, although presumably an I/O port could be the final destination of a message. In the first embodiment in Horst, however, the PEs are actually interconnected unidirectionally. In both cases the message blocks are of a fixed size and format (address, control information, data, etc.). In contrast, Applicants' I/O mechanism is shared by (i.e., connected to) all PEs.

But moreover, the architecture described in Horst is not a SIMD arrangement, as required by Applicants' claims. The passages that the Office refers to in column 1, lines 7-22 of Horst are merely by way of introduction to the different varieties of parallel processor architecture and do not state that Horst discloses a SIMD architecture comprising all the features specified in claim 23. Horst actually acknowledges later in column 1, at lines 56-64, that all of those known architectures contain problem areas that prior research has not solved. Horst therefore discloses a "new computer architecture" (see column 1, lines 67-68). Computation is performed by a set of tasks flowing through the network, as explained in column 2, lines 15-16. The architecture cannot possibly be SIMD, which requires a Single Instruction to be executable at each and every one of the processing elements (PEs) of the array. Instead of the I/O architecture specified in Applicants' claims, Horst operates by sending transmission packets between PEs, each such packet carrying the state of execution from the previous PE.

The Office is reminded that an anticipating reference must disclose *all* of the limitations defined by a claim, and with respect to claim 23, Horst is lacking a number of them.

Claims 24-31 and 58 variously depend from claim 23, and therefore incorporate all of the features of that base claim. Accordingly, for at least the foregoing reasons claims 23-31 and 58 are believed to be patentably distinguishable over the Horst patent. It is therefore respectfully requested that the rejection of these claims under 35 U.S.C. §§ 102(a) and 102(b) be withdrawn.



Claims 14, 15, 33, 36, 37, 42, 43, 48, 49, 54, and 55 again stand rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Brown (US Patent 5,872,993) in view of Childers et al. This rejection is respectfully traversed.

Independent claim 14 defines a data processing architecture comprising a parallel array of processing elements arranged in a single instruction multiple data processing array; a hardware accelerator unit operable to receive, in series, processing requests from the processing elements and to return processing results to respective processing elements when those processing results are available; and an input/output system which is operable to transfer respective processing requests from the processing elements to the hardware accelerator, and to return processing results to the processing elements concerned, wherein the processing elements are operable to process the returned processing results when all such results are returned, or after a predetermined time period.

Claim 15 differs from claim 14 only in that the accelerator is operable to return processing results to respective processing elements in the order in which processing requests were received by the accelerator.

Brown discloses an architecture in which a digital signal processor DSP 300 communicates with hardware accelerators HW ACC1 (330) and HW ACC2 (331) through a data flow processor DFP (320), which regulates the flow of requests and returns between the DSP and the HW ACCs. As pointed out by the examiner, column 7, lines 40-52, state that the HW ACCs use the memory of the DSP and that the DSP can be executing whilst the HW ACCs run using the DSP's memory as data buffers.

The obviousness rejection as presently framed is based on the assertion that it would be obvious to apply the SIMD architecture in Childers et al. to the Hardware Accelerator concept in Brown. The Office has previously rejected these same claims on the basis that it would be obvious to apply the Hardware Accelerator concept in Brown to the SIMD architecture taught in Childers et al. The Office has therefore approached the question of obviousness from both sides, using the same combination of prior art references, Brown and Childers et al. However, the Office has declined to make any comment, favorable or otherwise, in regard to Applicants' responses to the previous Office Actions, so that Applicants are uncertain which objection or objections the Office wishes to maintain. Nevertheless, in the interests of proceeding with the current issues, Applicants present the following points.

In a SIMD processor, it only makes sense to pass data from a parallel processor to a hardware accelerator if that accelerator is also parallel; otherwise there is a “serialization problem”. This is something to be avoided in parallel processors because after a data item from a PE has been sent to the accelerator, it would be necessary to wait for that item of data to be processed and returned before the next data item from another PE could be sent. The whole point of parallel processing would then be lost.

It is not at all apparent that one of ordinary skill in the art would have had a reasonable expectation of success in making the Office's proposed combination. The obstacles to combining Childers et al. and Brown, which have been overcome by the invention as set out in claims 14 and 15, can be summarized as follows:

First, a more complex accelerator is required which, although it may receive data sequentially, will have to be parallel internally (typically pipelined) so it can process several of the requests at once. At any point in time, there may be several data items queued up waiting to be processed by the accelerator, several being processed and several being returned to the PEs.

Second, in order to avoid inefficiencies for the SIMD processor, it must be able to do something else while this is happening. This requires (a) multi-threading and (b) an I/O mechanism that allows the data to be returned to the PE memory while the PEs are otherwise involved with something else, that is, effectively without their direct intervention. This is significantly more complex than a sequential processor where the data could just be returned to shared memory. On the contrary, the memory of PEs in a SIMD processor is typically “private” (to each PE) rather than shared.

Third, communication between the SIMD processor and the accelerator needs a mechanism to “tag” each data item with the identity of the originating PE so it can be returned to the correct place. In this regard it is important to note that because the SIMD processor will not return to processing the data returned by the accelerator until all the data has been returned (or there is a timeout) there is no need to re-order the data if it is returned out of order as long as it is returned to the correct PE. This alone constitutes a significant technical difference over the prior art.

It should therefore be apparent that, far from being a “simple” operation to combine the teachings of the two cited documents, it is actually a complex situation that raises as many problems as it “solves”. It leaves unanswered questions as to how to design

mechanisms by which the SIMD processor can interact with the accelerator in an effective and efficient way. Claims 14 and 15 are concerned with two particular ways of achieving this. In claim 14 the accelerator is operable to return processing results (from the accelerator) when those processing results are available. In claim 15, the accelerator is operable to return processing results to respective processing elements in the order in which processing requests were received by the accelerator. It is therefore submitted that neither of independent claims 14 and 15 is obvious in light of the Brown and Childers et al. documents.

Claims 33, 36, 37, 42, 43, 48, 49, 54, and 55 variously depend from claims 14 and 15, and consequently incorporate the same features as those defined by their respective base claims. It is therefore respectfully submitted that claims 14, 15, 33, 36, 37, 42, 43, 48, 49, 54, and 55 are patentably distinguishable over the Brown and Childers et al. patents regardless of whether these documents are considered individually or in any combination. Accordingly, it is respectfully requested that the rejection of these claims under 35 U.S.C. § 103(a) be withdrawn. It is noted that arguments put forth in previous responses have gone without comment from the Office. It is of course hoped that the Office will favorably consider the points set forth above. However, if the Office chooses to maintain this rejection, it is respectfully requested that it provide some basis, grounded on legal authority, for considering Applicants' arguments unpersuasive.

Claim 16 stands rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Childers et al. in view of Brown. This rejection is respectfully traversed.

Claim 16 depends from claim 1, and is therefore patentably distinguishable over Childers et al. for at least the reasons set forth above. It is noted that the Office acknowledges, in numbered paragraph 4, that "Childers did not specifically show the number of PEs was determined based on the size of the data packet as claimed."

Brown, too, fails to disclose an input device "operable to distribute whole data packets of potentially varying size across one or more processing elements such that the number of processing elements across which each whole data packet is distributed is dynamically determined based at least in part on the size of the whole data packet," and therefore fails to make up for the deficiencies of Childers et al. Because these features would be lacking in any combination of Childers et al. with Brown, such a combination would fail to support a *prima facie* case of obviousness.

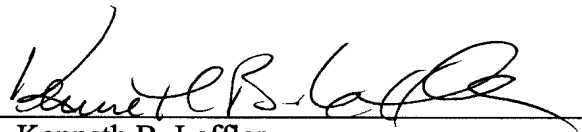
Moreover, as explained above, one of ordinary skill in the relevant art at the time of the invention would have found neither the necessary motivation nor the expectation of success in combining Childers et al. with Brown because of the many difficulties encountered in applying Brown's arrangement into a SIMD environment for which it was not intended.

For at least the foregoing reasons, claim 16 is believed to be patentably distinguishable over any combination of Childers et al. with Brown. Accordingly, it is respectfully requested that the rejection of claim 16 under 35 U.S.C. §103(a) be withdrawn.

The application is believed to be in condition for allowance. Prompt notice of same is respectfully requested.

Respectfully submitted,  
Potomac Patent Group PLLC

Date: June 30, 2006

By:   
Kenneth B. Leffler  
Registration No. 36,075

P.O. Box 270  
Fredericksburg, Virginia 22404  
703-718-8884